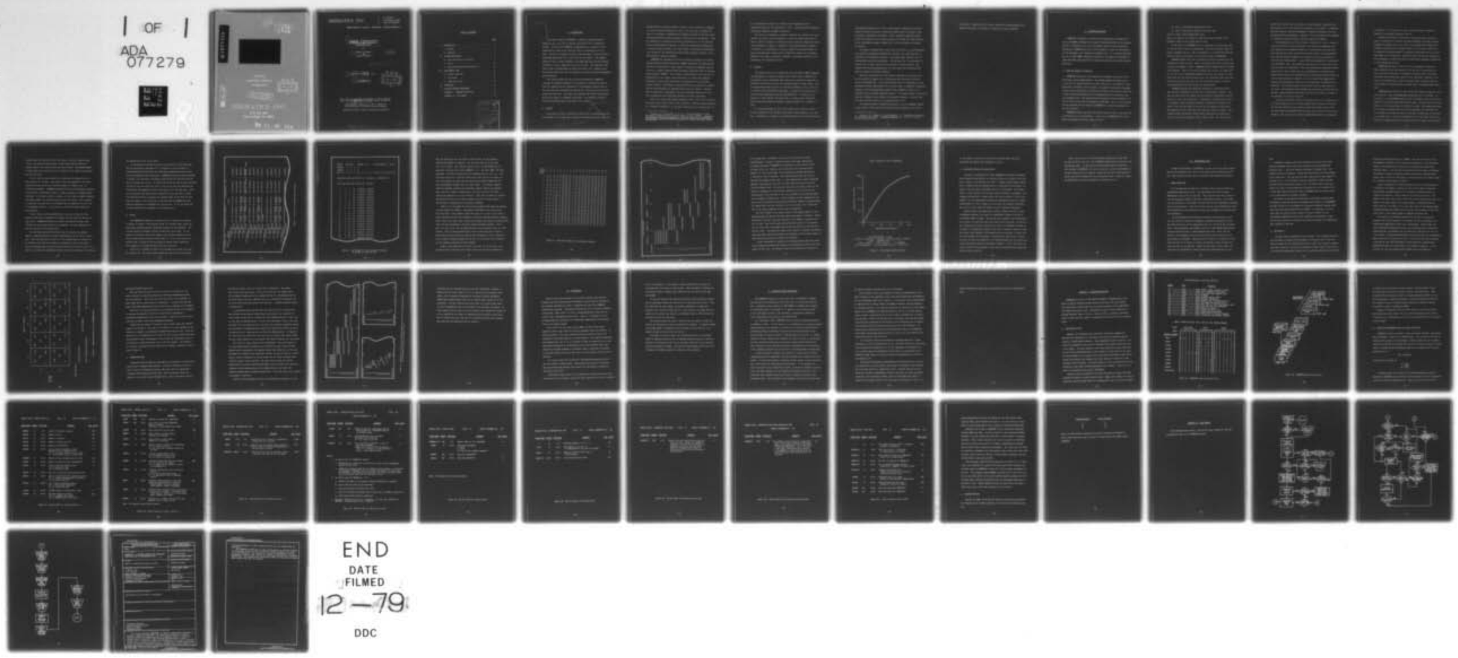
AD-A077 279 DESMATICS INC STATE COLLEGE PA F/G 5/10
SCHEDULER: A COMPUTER PROGRAM FOR SCHEDULING ADMINISTRATION OF --ETC(U)
NOV 79 R L GARDNER , D E SMITH N00014-79-C-0128
UNCLASSIFIED TR-112-3 NL

| OF |
ADA
077279



END
DATE
FILMED
12-79
DDC

(B)

— STATISTICS —

— OPERATIONS RESEARCH —

— MATHEMATICS —

DESMATICS, INC.

P.O. Box 618
State College, Pa. 16801

DESMATICS, INC.

P. O. Box 618
State College, Pa. 16801
Phone: (814) 238-9621

Applied Research in Statistics - Mathematics - Operations Research

6

SCHEDULER: A COMPUTER PROGRAM FOR SCHEDULING ADMINISTRATION OF PERFORMANCE TESTS

by

10

Robert L. Gardner
and
Dennis E. Smith

12 57

14 TR-112-3

9 TECHNICAL REPORT NO. 112-3

11 November 1979

This study was supported by the Office of Naval Research under Contract No. N00014-79-C-0128, Task No. NR 207-037

15

391 156

JUB

TABLE OF CONTENTS

I. INTRODUCTION

This report describes SCHEDULER, a computer program developed by Desmatics, Inc. as an aid in solving a particular type of scheduling problem. The fact that SCHEDULER is implemented on a computer is only significant in terms of the speed with which a scheduling run can be made. The use of a computer performs no special magic in this particular scheduling application, nor in any other for that matter. The computer program is only a formal statement of an algorithm which has been devised to perform a task in compliance with a set of rules. The algorithm could just as well be accomplished with paper and pencil, but a computer does the job much faster and, once the program is properly tested and "debugged", more accurately.

This report defines the type of problem addressed by SCHEDULER, indicates the approach employed by the program in attacking the problem, lists the capabilities and limitations of the program, discusses some of the runs which have been made and presents the conclusions which have been drawn from these runs. The report also suggests some further developments which would enhance the program's efficiency. The appendices provide a technical description and a flow chart of the program.

A. PROBLEM

There exists a type of problem, not unlike that of scheduling the work for machines in a job shop, which confronts experimenters who must plan the administration of research studies in which a pool of subjects is assigned to a set of related experiments or tests. For this type of problem, the task is to determine a schedule which minimizes the span of time over which testing takes place, while satisfying a specified set of constraints. Thus, the problem is to provide the efficient utilization of a set of resources. Given a varying group of subjects, a relatively complicated configuration of resources and facilities, and a large number of interrelated experiments to be conducted, the scheduling problem becomes extremely difficult for an experimenter to undertake unaided.

SCHEDULER was developed as an aid in preparing schedules at the Naval Aerospace Medical Research Laboratory (NAMRL) Detachment for human subjects used in the evaluation of a large number of candidate performance tests as a part of a research program called Performance Evaluation Test for Environmental Research (PETER).[1] This research program requires that each test be administered to each subject once each day for 15 consecutive workdays. Sequencing of tests within types must be controlled due to anticipated practice effects. Furthermore, testing facilities are limited as to number, capacity and flexibility in changing from one test to another. Subjects must be scheduled in conformance with rules governing sequencing of tests and in compliance with constraints imposed by facility limitations.

There are a great many possible schedules, some of which are of superior efficiency, which comply with all the constraints. However, manual production of a "feasible" schedule (i.e., one which complies with all constraints, but

---

[1] Kennedy, R.S. and Bittner, A. C., Jr. "The Development of a Navy Performance Evaluation Test for Environmental Research (PETER)" Symposium Proceedings Productivity Enhancement: Personnel Performance Assessment in Navy Systems, October 1977, Navy Personnel R & D Center, San Diego, Calif.

is not necessarily optimal) for problems of this magnitude takes a significant amount of the experimenter's time. Constructing and evaluating alternative schedules is almost prohibitive.

Not only is it necessary to prepare a schedule for a given set of conditions, but also it may be necessary to revise the schedule from time to time to reflect the impact of unforeseen factors which may temporarily alter the availability of subjects, equipment or test administration personnel. In addition, it is desirable to be able to assess the potential impact of policy decisions which could result in significant long-term or permanent changes in the number of subjects, equipment or personnel available to the performance test evaluation project.

B. APPROACH

The dynamic nature of the scheduling task confronted by NAMRL suggested the development of a computer program which could be used to simulate the environment in which scheduling takes place and to provide an algorithm of flexibility sufficient to cover the anticipated range of input values. Such a program should permit the user to construct schedules for the current set of resources using alternative strategies, to evaluate each strategy and to select the most satisfactory one for implementation. It should then be possible to input specific real data (even to include subject names and specific dates and times) and produce a working schedule for a segment of time.

As testing proceeds, new runs of the program can be made to produce further increments of the schedule which reflect actual progress up to that date. Furthermore, it would be of considerable benefit from an administrative and planning standpoint to be able to make computer simulation runs which assess the effects on project completion brought about by the gain or loss of subjects, equipment or personnel. Such a simulation capability would permit the executive user to answer questions concerning the expected impact of any contemplated resource changes and to evolve strategies to optimize the results.

The computer program which evolved from this effort is an inplementation of a heuristic scheduling algorithm that provides, for a specified set of conditions, a deterministic solution. The program logic embodies a set of user-specified rules and constraints. For any given run it is necessary to provide a set of input data and fix the values of certain options. The program then proceeds systematically on a day-by-day and period-by-period basis to construct a feasible schedule.

There are indications that this heuristic approach is a practical one for the relatively complex set of interacting rules involved in this particular scheduling task. It has been observed that it is impossible to find optimized solutions analytically for all except small or specially constructed problems, since most scheduling problems of any complexity belong to the insoluble class of problem called NP-complete. "For these reasons practitioners often are willing to settle merely for a feasible schedule, so long as they have some indication that it is a reasonably good one."[1]

The approach adopted by Desmatics has been to use the SCHEDULER program to make several runs in which the input and control parameter options have

---

[1] Garey, M. R., Graham, R. L. and Johnson, D. S. "Performance Guarantees For Scheduling Algorithms," Operations Research, Vol. 26, No. 1, January-February 1978, pp. 3-21.

been varied. Comparison of the results provides an understanding of how schedule efficiency is affected by variations in input parameters.

II. PROGRAM DESCRIPTION

SCHEDULER is designed to construct test administration schedules for use in the NAMRL performance test research program described in the previous section. Although its use is not necessarily limited to this particular application, SCHEDULER was specifically configured to meet the requirements of the performance test evaluation scheduling task. It is readily adaptable for use in other NAMRL scheduling requirements, including the scheduling of human and human analog subjects for impact acceleration and vibration experiment runs.

A. INPUT AND THEORY OF OPERATION

SCHEDULER provides for the production of schedules in which up to ten tests may be given simultaneously. The program assumes that there are up to ten sets of testing facilities in which tests are to be scheduled. These facilities, referred to here as "labs", are assumed to contain the equipment (chairs, desks, projectors, tachistoscopic displays, rotary pursuit apparatus, or other facilities) necessary for the administration of a particular type of performance test. Test administration personnel are also considered to be part of each lab facility for purposes of scheduling. That is, test administration personnel are not treated by SCHEDULER as a resource independent of the lab facilities.

SCHEDULER assumes that there are distinct types of tests, each requiring a particular set of lab facilities. In all runs of SCHEDULER made to date there were assumed to be three types of tests:

(a) Type 1, individually administered tests,

(b) Type 2, small-group (apparatus limited) tests

and (c) Type 3, group administered tests.

There is no limit to the number of types which may be defined, but as mentioned, there may be no more than ten labs.

At the start of a SCHEDULER run it is necessary to specify what type of test each lab is set up to accommodate. This lab setup must remain in effect throughout the run. This, incidentally, is a restriction which may possibly limit the efficiency of the current version of SCHEDULER.

SCHEDULER permits the user to establish a priority hierarchy for consideration of types of tests. This is achieved by the order in which types of tests are assigned to labs. SCHEDULER always considers labs in the same fixed sequence. Thus, if the user specifies that the first of three labs is to be set up for individually-administered tests, the second for small group tests and the third for group tests, then individual tests will have first priority and group tests will have last priority when it comes to allocating subjects to tests during the scheduling run.

SCHEDULER performs the scheduling operation on a day-by-day basis. Each day may be divided by the user into a fixed number of periods. The user may also divide the periods into two blocks by specifying the period which is to start the second block. Scheduling is performed with respect to "workdays" within an internal calendar, which must be defined at the start of each run and linked to the real world calendar by specifying a particular start date, which may be in the past, present or future. The program assumes that Monday through Friday constitutes the normal work week, but the user may specify another range of days. The user may also specify that certain days are holidays, and may designate certain days on which some or all of the subjects are expected to be on leave. If holidays and leave days are carefully designated, a realistic estimate of the completion date may be determined. However, if only the length of the schedule in workdays is desired, holidays and leave days may be ignored.

Two additional types of input data are required for a run: a table of tests and a roster of subject personnel. The test table designates every test required to be scheduled during the run and specifies the test type. Each test may have other tests within the test table as its prerequisites. Up to 120 tests may be included in the table but no more than twenty of these may have prerequisites. A test which has prerequisites is referred to as a "milestone" test. Each milestone test may have up to ten other tests from the table as prerequisites. The rule governing the scheduling of milestone tests is that a subject may not be scheduled for a milestone test administration until he has completed all of the required administrations of all of its prerequisites. However, it is not required that <u>all</u> subjects complete all prerequisites before the first subject may be scheduled for a milestone. The user may specify for each test a date on which that test may start. The program will not schedule any test until its start date has arrived. Unless otherwise specified, each test in the table is assumed to be available at the start of the run.

The user must provide a roster of subjects who are to be scheduled for all tests during the run. Up to 100 subjects may be listed in the roster. The user may specify for each subject an entry date and/or an exit date. No subject will be scheduled before his entry date nor after his exit date. (However, exit dates are not used in establishing priority for scheduling of subjects.) In the absence of entry and/or exit dates a subject is assumed to be available throughout the run.

Scheduling proceeds on a day-by-day, period-by-period, lab-by-lab basis. At the start of each day the program examines the roster, and the calendar for several days in the future, to determine which subjects are available currently and for enough workdays in the future to allow them to complete the required number of consecutive administrations of any test which might start on that day. Indicators within the computer program are set accordingly to obviate the need for repeatedly determining availability for each period and each lab.

SCHEDULER then tries to find a test to be scheduled in each lab. Tests scheduled in the first period of the day (or the first period of the second block of periods, if blocking is allowed) cannot be replaced with other tests during later periods of the same block. However, if no test can be scheduled due to lack of subjects (busy or done) during the first period of a block, one may be started later if subjects become available.

SCHEDULER gives priority to tests which have already started over tests which have not. This is accomplished through the use of two queues. At the start of a run SCHEDULER loads a queue of startable tests with the test numbers of the first N tests from the test table. The user must specify the number, N, of tests which may be active at one time. The program then looks in the queue of started tests for a test to schedule in a particular lab. If a suitable test cannot be found in this queue it looks next in the queue of startable tests. If a test is found which is of the right type for the lab, and if enough subjects can be found, the test is added to the queue of started tests and removed from the other queue. Since the program looks first in the queue of started tests, it effectively gives priority to started tests in the order in which they were started. The maximum number of tests which can be active at once is limited by the length of the queues, which is 120 tests.

After a test has been selected for scheduling, the program tries to find subjects to be tested. Each subject must be scheduled for one administration per day on each of a specified number of workdays (e.g., 15 consecutive workdays). SCHEDULER accomplishes this by maintaining a subjects-by-tests table called LOGG in which it counts the number of administrations completed. If an entry is zero or greater, but less than the number required, the subject needs to be scheduled for the test in question. Once a subject is scheduled, his LOGG entry is incremented and set negative so he will not be rescheduled that day. At the start of a new day all LOGG entries are reset positive.

After a subject has been scheduled for a test, he is busy for that period and must not be scheduled for another test in another lab during the same period. SCHEDULER controls the conflict problems by setting a busy indicator for each subject after he is scheduled. All busy indicators are reset at the start of the next period.

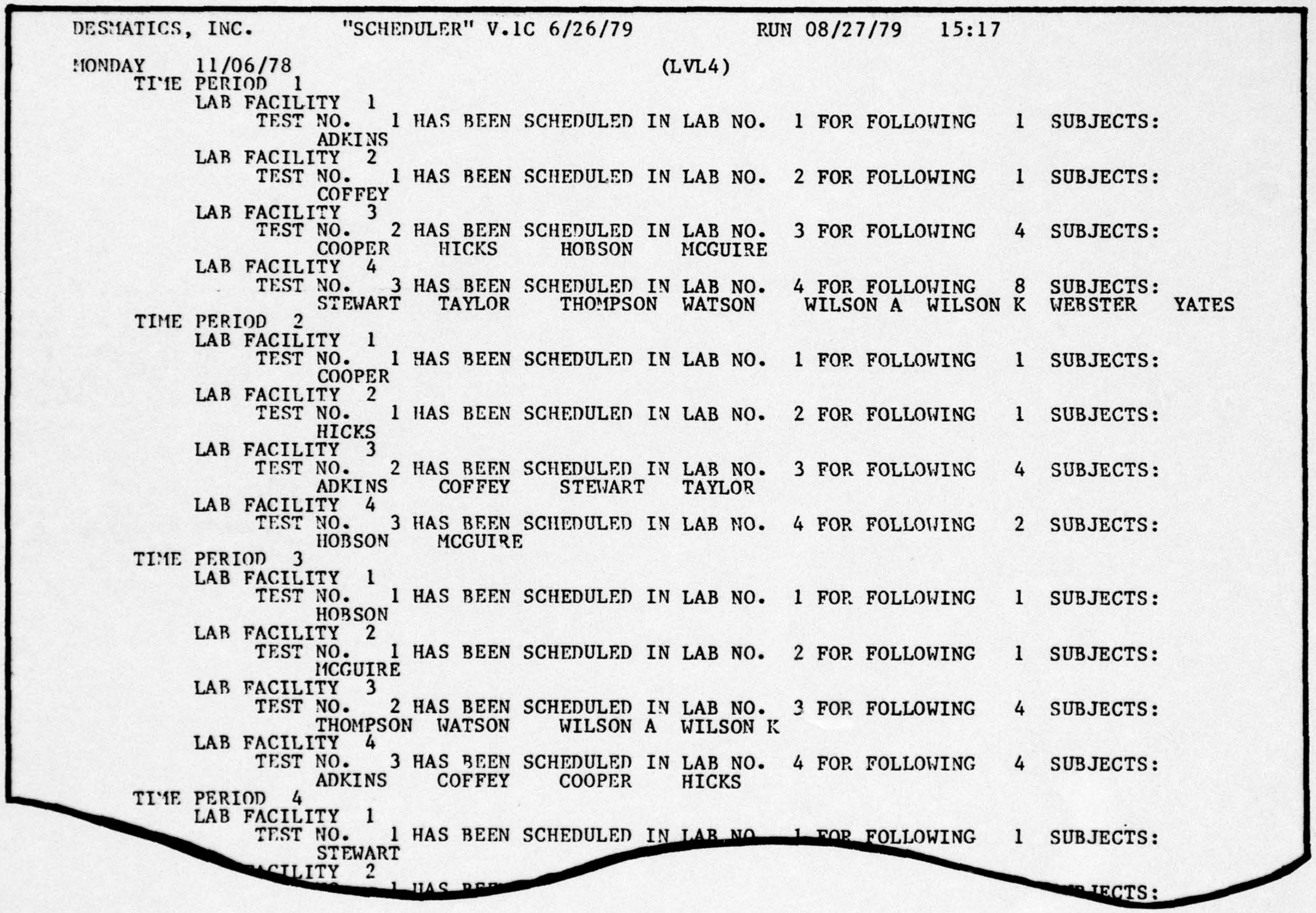
The previously mentioned test table also must specify the maximum number of subjects which can be tested simultaneously in one lab for a given test. This table may also be used to set the minimum number of subjects to be scheduled, which may be the same as the maximum or some lesser number. Using the minimum group size option and setting the size equal to one for all tests is an alternative scheduling strategy which may be compared with requiring the maximum group size in all cases.

It is possible to specify that two or more labs are of the same type. This is particularly applicable if it is desired to see the effect of providing additional facilities for individually administered tests, which are the bottleneck in most situations. SCHEDULER provides an option intended to improve the efficiency of schedules produced when two labs have been set up for the same type of test. To apply this option the user not only specifies the type of test for which each lab is to be set up, but also indicates the "dual-track" relationship of the two labs which are of the same type. Without this feature both labs would be scheduled for the same test, but when this dual-track option is applied, the program checks at the start of each block of periods to see if one lab of the pair could accommodate all the subjects remaining to be scheduled for a given test. If so, the second lab is then available to be scheduled for a different test.

B. OUTPUT

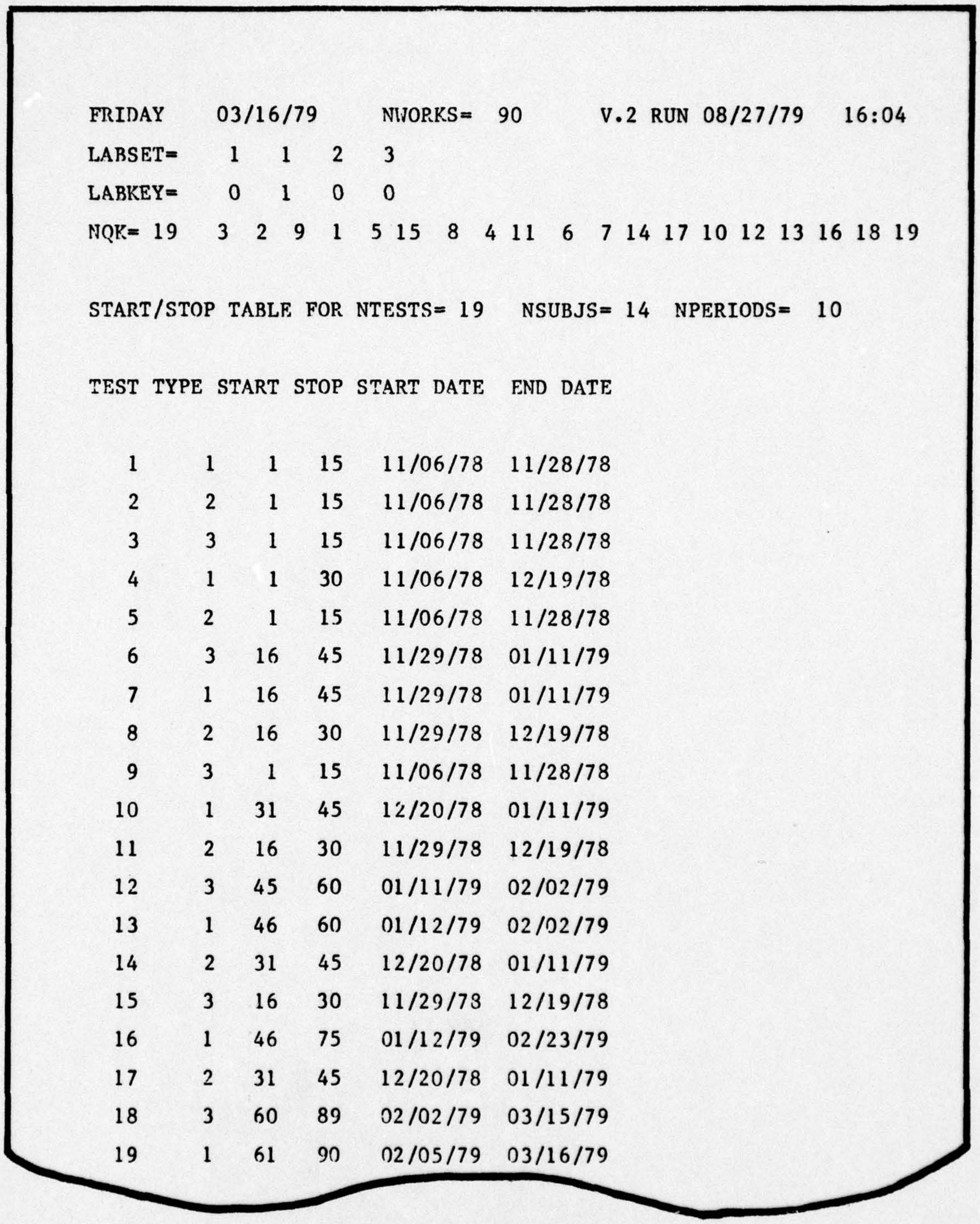
The SCHEDULER program as it currently exists is capable of producing a variety of outputs. The principal output is a printout which traces the day-by-day, period-by-period, lab-by-lab operation of the algorithm. The user may specify the degree of detail produced for this output. At one extreme it provides a list of the names of the subjects scheduled in each lab for each period of every day (see Figure 1). At the other extreme it produces only a listing of the user-selected options (input conditions) and a summary of the final results at the end of the run.

Figure 2 is an example of some of the information provided at the end of a typical run. The upper portion summarizes some of the run conditions.



```
DESMATICS, INC.       "SCHEDULER" V.1C 6/26/79          RUN 08/27/79   15:17

MONDAY    11/06/78                             (LVL4)
     TIME PERIOD  1
          LAB FACILITY  1
               TEST NO.   1 HAS BEEN SCHEDULED IN LAB NO.  1 FOR FOLLOWING   1  SUBJECTS:
                    ADKINS
          LAB FACILITY  2
               TEST NO.   1 HAS BEEN SCHEDULED IN LAB NO.  2 FOR FOLLOWING   1  SUBJECTS:
                    COFFEY
          LAB FACILITY  3
               TEST NO.   2 HAS BEEN SCHEDULED IN LAB NO.  3 FOR FOLLOWING   4  SUBJECTS:
                    COOPER    HICKS     HOBSON    MCGUIRE
          LAB FACILITY  4
               TEST NO.   3 HAS BEEN SCHEDULED IN LAB NO.  4 FOR FOLLOWING   8  SUBJECTS:
                    STEWART   TAYLOR    THOMPSON  WATSON    WILSON A  WILSON K  WEBSTER   YATES
     TIME PERIOD  2
          LAB FACILITY  1
               TEST NO.   1 HAS BEEN SCHEDULED IN LAB NO.  1 FOR FOLLOWING   1  SUBJECTS:
                    COOPER
          LAB FACILITY  2
               TEST NO.   1 HAS BEEN SCHEDULED IN LAB NO.  2 FOR FOLLOWING   1  SUBJECTS:
                    HICKS
          LAB FACILITY  3
               TEST NO.   2 HAS BEEN SCHEDULED IN LAB NO.  3 FOR FOLLOWING   4  SUBJECTS:
                    ADKINS    COFFEY    STEWART   TAYLOR
          LAB FACILITY  4
               TEST NO.   3 HAS BEEN SCHEDULED IN LAB NO.  4 FOR FOLLOWING   2  SUBJECTS:
                    HOBSON    MCGUIRE
     TIME PERIOD  3
          LAB FACILITY  1
               TEST NO.   1 HAS BEEN SCHEDULED IN LAB NO.  1 FOR FOLLOWING   1  SUBJECTS:
                    HOBSON
          LAB FACILITY  2
               TEST NO.   1 HAS BEEN SCHEDULED IN LAB NO.  2 FOR FOLLOWING   1  SUBJECTS:
                    MCGUIRE
          LAB FACILITY  3
               TEST NO.   2 HAS BEEN SCHEDULED IN LAB NO.  3 FOR FOLLOWING   4  SUBJECTS:
                    THOMPSON  WATSON    WILSON A  WILSON K
          LAB FACILITY  4
               TEST NO.   3 HAS BEEN SCHEDULED IN LAB NO.  4 FOR FOLLOWING   4  SUBJECTS:
                    ADKINS    COFFEY    COOPER    HICKS
     TIME PERIOD  4
          LAB FACILITY  1
               TEST NO.   1 HAS BEEN SCHEDULED IN LAB NO.  1 FOR FOLLOWING   1  SUBJECTS:
                    STEWART
            ACILITY  2
                    1 HAS BEE                                                  JECTS:
```

Figure 1: Portion of a Detailed Daily Schedule



```
FRIDAY     03/16/79      NWORKS=  90       V.2 RUN 08/27/79   16:04

LABSET=    1   1   2   3

LABKEY=    0   1   0   0

NQK= 19    3   2   9   1   5 15   8   4 11   6   7 14 17 10 12 13 16 18 19


START/STOP TABLE FOR NTESTS= 19   NSUBJS= 14  NPERIODS=  10


TEST TYPE START STOP START DATE  END DATE


   1     1     1   15   11/06/78  11/28/78
   2     2     1   15   11/06/78  11/28/78
   3     3     1   15   11/06/78  11/28/78
   4     1     1   30   11/06/78  12/19/78
   5     2     1   15   11/06/78  11/28/78
   6     3    16   45   11/29/78  01/11/79
   7     1    16   45   11/29/78  01/11/79
   8     2    16   30   11/29/78  12/19/78
   9     3     1   15   11/06/78  11/28/78
  10     1    31   45   12/20/78  01/11/79
  11     2    16   30   11/29/78  12/19/78
  12     3    45   60   01/11/79  02/02/79
  13     1    46   60   01/12/79  02/02/79
  14     2    31   45   12/20/78  01/11/79
  15     3    16   30   11/29/78  12/19/78
  16     1    46   75   01/12/79  02/23/79
  17     2    31   45   12/20/78  01/11/79
  18     3    60   89   02/02/79  03/15/79
  19     1    61   90   02/05/79  03/16/79
```

Figure 2: An Example of Some Information Provided at the End of a Typical Run

Here the completion date was Friday, 16 March 1979, and the schedule required 90 workdays to complete. The four labs were set up for type (1, 1, 2, 3) tests. Lab 2 was the second track of a dual-track pair, of which lab 1 was the first track (LABKEY(2) = 1). All tests (NQK = 19) were completed. The order in which they were completed was (3, 2, 9, 1,..., 19). The lower portion is a start/stop table which lists each of the tests, its type, the workdays on which each was started and stopped (completed) and the corresponding calendar dates for these events. In contrast, Figure 3 shows the workdays on which each individual subject started and completed each test [e.g., subject 1 started test 1 (a type 1 test) on the first workday and completed it on the fifteenth]. This table is useful in confirming compliance with the rule which requires that once a subject starts a test he must be scheduled for N consecutive workdays.

Figure 4 is a bar graph for tests versus workdays which shows the phasing of tests within the schedule. The numbers which make up the bars are the test type codes. Thus, Figure 4 shows that test 1 is a type 1 test (individually administered) which was scheduled on the first through fifteenth workdays. Tests 2 and 3 are types 2 and 3 respectively (small group and group administered), which also were scheduled on the first through fifteenth workdays. Test 4 also is a type 1 test, but its administration was not completed until the 30th workday. Just how this happened may be seen by referring back to Figure 3, which shows that the first five subjects were scheduled for test 4 during the first through fifteenth workdays and the rest of the subjects were scheduled for the sixteenth through thirtieth workdays.

Figure 4 also shows that a total of six tests, two of each type, were scheduled each of the first thirty days, while only five were scheduled the

```
TEST>  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
TYPE>  1  2  3  1  2  3  1  2  3  1  2  3  1  2  3  1  2  3  1
SUBJ
   1   1  1  1  1  1 16 16 16  1 31 16 45 46 31 16 46 31 60 61
      15 15 15 15 15 30 30 30 15 45 30 59 60 45 30 60 45 74 75
   2   1  1  1  1  1 16 16 16  1 31 16 45 46 31 16 46 31 60 61
      15 15 15 15 15 30 30 30 15 45 30 59 60 45 30 60 45 74 75
   3   1  1  1  1  1 16 16 16  1 31 16 45 46 31 16 46 31 60 61
      15 15 15 15 15 30 30 30 15 45 30 59 60 45 30 60 45 74 75
   4   1  1  1  1  1 16 16 16  1 31 16 45 46 31 16 46 31 60 61
      15 15 15 15 15 30 30 30 15 45 30 59 60 45 30 60 45 74 75
   5   1  1  1  1  1 16 16 16  1 31 16 45 46 31 16 46 31 61 61
      15 15 15 15 15 30 30 30 15 45 30 59 60 45 30 60 45 75 75
   6   1  1  1 16  1 30 16 16  1 31 16 45 46 31 16 61 31 75 61
      15 15 15 30 15 44 30 30 15 45 30 59 60 45 30 75 45 89 75
   7   1  1  1 16  1 30 16 16  1 31 16 45 46 31 16 61 31 75 61
      15 15 15 30 15 44 30 30 15 45 30 59 60 45 30 75 45 89 75
   8   1  1  1 16  1 30 16 16  1 31 16 45 46 31 16 61 31 75 61
      15 15 15 30 15 44 30 30 15 45 30 59 60 45 30 75 45 89 75
   9   1  1  1 16  1 30 16 16  1 31 16 45 46 31 16 61 31 75 61
      15 15 15 30 15 44 30 30 15 45 30 59 60 45 30 75 45 89 75
  10   1  1  1 16  1 30 16 16  1 31 16 45 46 31 16 61 31 75 61
      15 15 15 30 15 44 30 30 15 45 30 59 60 45 30 75 45 89 75
  11   1  1  1 16  1 30 31 16  1 31 16 45 46 31 16 61 31 75 76
      15 15 15 30 15 44 45 30 15 45 30 59 60 45 30 75 45 89 90
  12   1  1  1 16  1 30 31 16  1 31 16 45 46 31 16 61 31 75 76
      15 15 15 30 15 44 45 30 15 45 30 59 60 45 30 75 45 89 90
  13   1  1  1 16  1 30 31 16  1 31 16 45 46 31 16 61 31 75 76
      15 15 15 30 15 44 45 30 15 45 30 59 60 45 30 75 45 89 90
  14   1  1  1 16  1 31 31 16  1 31 16 46 46 31 16 61 31 75 76
      15 15 15 30 15 45 45 30 15 45 30 60 60 45 30 75 45 89 90
```

Figure 3: Start/Stop Workdays for Individual Subjects

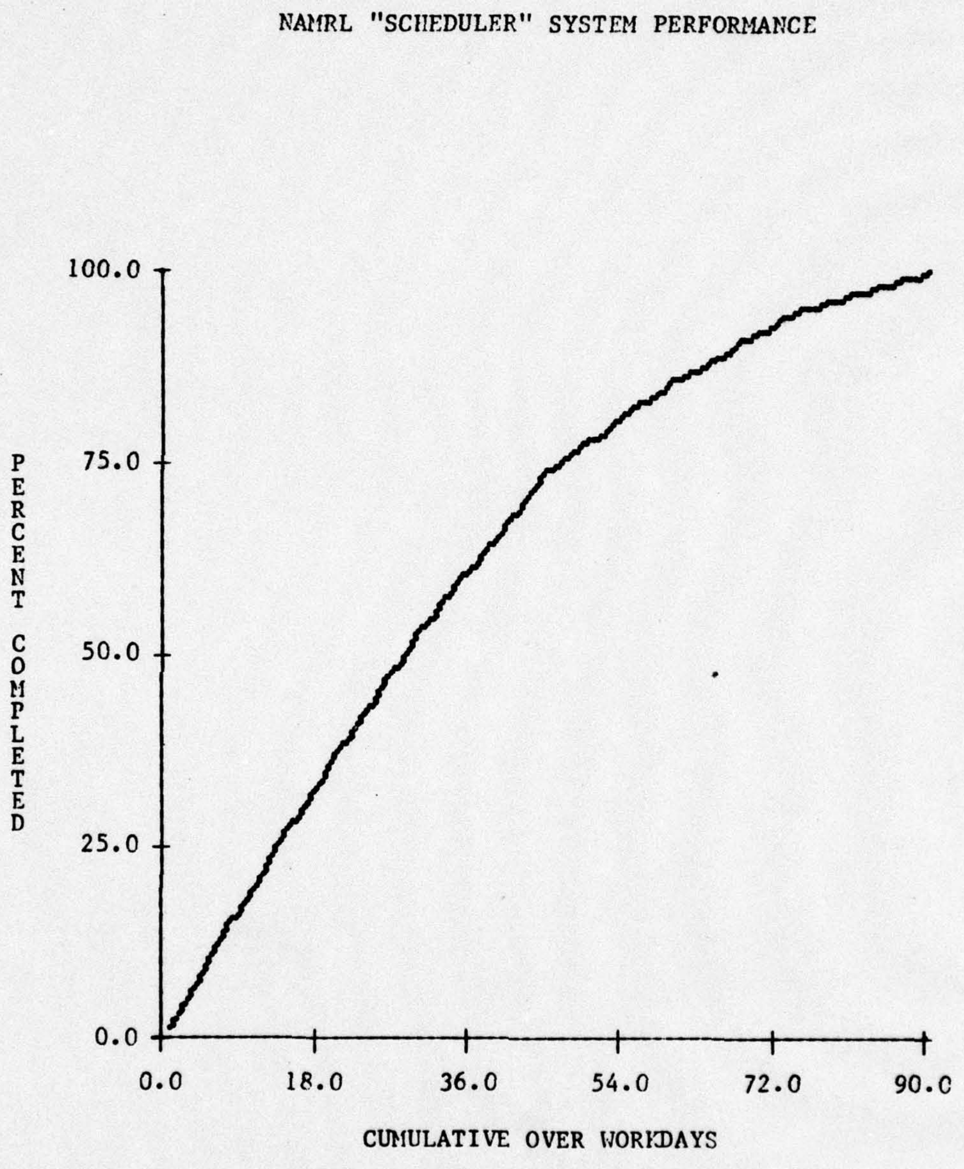
```
                    TEST START/STOPS

          NTESTS= 19     NSUBJS= 14     NLABYS= 4     NPERIODS= 10

TEST/DAYS=>            15             30             45             60             75             90

    1     111111111111111
    2     222222222222222
    3     333333333333333
    4     111111111111111111111111111111
    5     222222222222222
    6                    333333333333333333333333333333
    7                    111111111111111111111111111111
    8                    222222222222222
    9     333333333333333
   10                                   111111111111111
   11                    222222222222222
   12                                                  333333333333333
   13                                                  111111111111111
   14                                   222222222222222
   15                    333333333333333
   16                                                  111111111111111111111111111111
   17                                   222222222222222
   18                                                                 333333333333333333333333333333
   19                                                                 111111111111111111111111111111

TEST/DAYS=>            15             30             45             60             75             90
```

Figure 4: Test Phasing Bar Graph

31st through 45th. Thereafter, only two or three tests were active simultaneously. Because of computer printer line length limitations, bar graphs produced by SCHEDULER are restricted to cover only the first 120 days of any run. However, only 90 days have been reproduced here. Figure 5 is a plot which shows the percentage of schedule completion as a function of time in workdays. The legend indicates when the run was made and the program version used. It also summarizes the input conditions employed, including the test type for which each lab was set up and the priority sequence used. The start and stop dates are shown, as well as the number of workdays and the percentage completion at the end of the run. "Utilization" is the fraction of the total available lab periods (periods x labs x workdays) actually used. Had the schedule not been completed at the end of the run (as when the run exceeds the allowed amount of computer time), the legend would also have included a list of the tests still active and a list of the tests completed, in the order of completion.

The item "10 MAX" in the legend refers to the maximum number of total administrations per day for all tests which each subject is allowed. Unless specified by the user, this value defaults to the number of periods there are in a day. Thus in this case each subject was allowed to be scheduled once in each of the ten periods of every day (but only once a day for each specific test). The end-of-run output also includes a frequency distribution table (not shown here) which provides an indication of the average number of administrations per day each subject was scheduled.

If more information is required, the program produces detailed printed output showing precisely which subjects were scheduled in each lab for each period of every day. The detailed output also indicates the contents of



```
                  V.1C RUN 08/27/79    16:04
19 TESTS, 14 SUBJECTS, 10 PERIODS, 15 ADMINS,10 MAX, 2 BLOCK(S)
                  4 LABS,     PRIORITY SEQ:  1  1  2  3
     START: 11/06/78       STOP: 03/16/79        90 WORKDAYS
  0.317 UTILIZATION      44.3 AVG ADMINS PER DAY    100.0% DONE
```

Figure 5: Percentage Completion Graph

the two queues at the end of each period and shows which tests were considered when making test assignments to labs.

C. ADDITIONAL OPTIONS AND CAPABILITIES

An option is provided which allows SCHEDULER to simulate the random, short-duration unavailability of individual subjects, as might be the case when a subject is on sickcall for a day. A similar option permits simulation of random short-term unavailability of individual lab facilities, as might be the case when an equipment failure is experienced. To activate either of these options, the user must supply probability thresholds. The program then generates a random number each time a subject or lab is considered. If the random number exceeds the threshold specified for subject unavailability or for an outage in a particular lab (separate thresholds may be set for each lab), the subject or lab is bypassed temporarily.

Another option, the dump/restart feature, permits the user to cause intermediate results to be output to intermediate storage at the end of a specified number of weeks. The run may then continue to the completion of the schedule, at which time the final results will be printed. By specifying a high level of detail, the user can cause the intermediate results to be output to intermediate storage at the end of a specified number of weeks. The run may then continue to the completion of the schedule, at which time the final results will be printed. By specifying a high level of detail, the user can cause the production of a working schedule for the period prior to the interruption point, as was shown in Figure 1. After this point the level of detail can be reduced so that further printing will be suppressed until the end of the run.

Later, as the end of the working schedule approaches in real time, the user can make a new run of the SCHEDULER program which can start at the interruption point. If the actual testing progress does not match the plan produced by SCHEDULER, the user can make any necessary adjustments in the data which was put in temporary storage and use this as the basis for another run which continues from the interruption point. Items not requiring adjustment can be used unchanged in restarting the run.

III. EXPERIMENTAL RUNS

During the development of SCHEDULER, a series of runs was made in which many of the parameters were fixed in value while others varied from run to run. This section describes the run conditions and discusses the results.

A. COMMON CONDITIONS

All runs employed the same set of nineteen tests, seven of which were individually-administered tests. Six small-group tests and six group-administered tests were also included. These nineteen tests were listed in a fixed rotational sequence (individual, small-group, group). Every sixth test was defined as a "milestone" test which had the five immediately preceding tests as prerequisites. Since every third test was a group-administered test, every milestone test was thus also a group test, although not all group tests were milestones.

All runs employed four laboratory facilities with the labs set up so that the first two could each accommodate individually-administered tests. The third lab was set up for small-group tests and the fourth lab for group tests. This essentially gave highest priority to individually-administered tests and lowest priority to group-administered tests. Every test was allowed to be scheduled for as few as one subject after trying to get as many subjects as possible. Subjects were not limited in the number of tests taken per day, but of course no subject was allowed to take the same test more than once in the same day. Except as noted later, every subject was required to be scheduled for fifteen consecutive administrations of each test.

A maximum of eighteen tests were allowed to be active at one time; thus the nineteenth test could not be started until all subjects had completed test 1. The first workday was Monday, 6 November 1978. The longest run took 135 workdays. Within this period there were six holidays but no leave days. The runs that were made always used an even number of periods per day. The "blocking" option was employed to divide the days into two time period blocks of equal size. The "dual-track" option was used in all runs to allow scheduling a different test in the second lab if it were found that the first lab could accommodate all the subjects still to be tested within the periods remaining.

The choice of the specific conditions outlined above reflects some conclusions evolved during early development and testing of the SCHEDULER program, as well as the development of additional capability options which have since been incorporated into the program. For instance, it is now clear that the better priority scheme is to give preference to the individually-administered tests which constitute the bottleneck, rather than to give priority to the group tests which are capable of accommodating more subjects at one time.

B. RUN RESULTS

Ten major runs are described in this section. The principal variables are the number of subjects and the number of periods per day. Runs were made for eight and ten periods per day and for nine, ten, fourteen, eighteen and twenty subjects. The runs with eighteen or twenty subjects and eight or ten periods correspond most closely to the conditions under which performance testing is now being carried out at NAMRL. Runs were also made for the same numbers of periods, but for half as many subjects, to see what might result if the subjects were divided into two groups and tested at two different times. A run with fourteen subjects was also made to facilitate interpolation between the nine-ten subject runs and the eighteen-twenty subject runs, in hope of identifying a clear break point in schedule length as a function of the number of subjects.

Figure 6 summarizes the results obtained for the runs undertaken, as a function of the number of periods per day and the number of subjects. The principal evaluation measure is the number of workdays required to complete the schedule. This is shown in the center of each cell. The number in the upper left corner of each cell is the utilization factor, the number in the upper right is the average number of administrations per subject per day, the number in the lower left corner is the maximum number of administrations any subject had in any day, and the number in the lower right corner is the average number of administrations per day.

The length of the schedules produced for the sets of conditions examined here is most sensitive to the manner in which scheduling of the individually-administered tests can be accomplished. Because there are seven individually-administered tests, each requiring fifteen consecutive administrations, the schedule will of necessity be 105 days long (15 x 7) if the number of subjects and periods per day allow exactly one individually-administered test to be given at a time. Such is the case when the number of periods per day times the number of labs dedicated to individually-administered tests is at least equal to the number of subjects. In ten periods, two labs can process twenty subjects per day through one

		Number of Subjects				
		9	10	14	18	20
Periods Per Day	8	.475 4.76	.396 2.71	.339 2.71	.264 2.11	.264 2.11
		90	90	105	135	135
		6 42.8	6 31.7	5 38.0	4 38.0	4 42.2
	10	.654 5.81	.475 .475	.317 3.16	.271 2.72	.271 2.72
		60	60	90	105	105
		6 52.3	6 47.5	6 44.3	5 48.9	5 54.3

Table Entries:

Utilization Factor		Ave. Administrations/Subject/Day
	No. of Workdays	
Max. Administrations/Subject		Ave. Administrations/Day

Figure 6: Summarized Results of Ten SCHEDULER Runs

individually-administered test.

When the number of subjects is ten or less it is possible for two labs to process all subjects through two individually-administered tests per ten-period day. Since there are seven such tests to be scheduled, it takes sixty days to process nine or ten subjects. It should be noted that eight individually-administered tests could also be completed in 60 workdays.

When there are fourteen subjects and ten periods per day it is possible to complete the schedule in 90 days because partial overlapping of individually-administered tests can be achieved. This may be seen by referring back to Figure 3 and its accompanying discussion.

Restricting the number of periods per day to eight causes the schedule length to be extended. With nine or ten subjects it is not possible for all subjects to complete two individually-administered tests per day. While the majority can, the remainder must be held over into the next fifteen-day span. The resultant partial overlapping of tests causes the scheduling of nine or ten subjects in eight periods per day to take 90 workdays. The schedules for fourteen, eighteen and twenty subjects are similarly extended, as may be seen in Figure 6.

C. ABBREVIATED RUNS

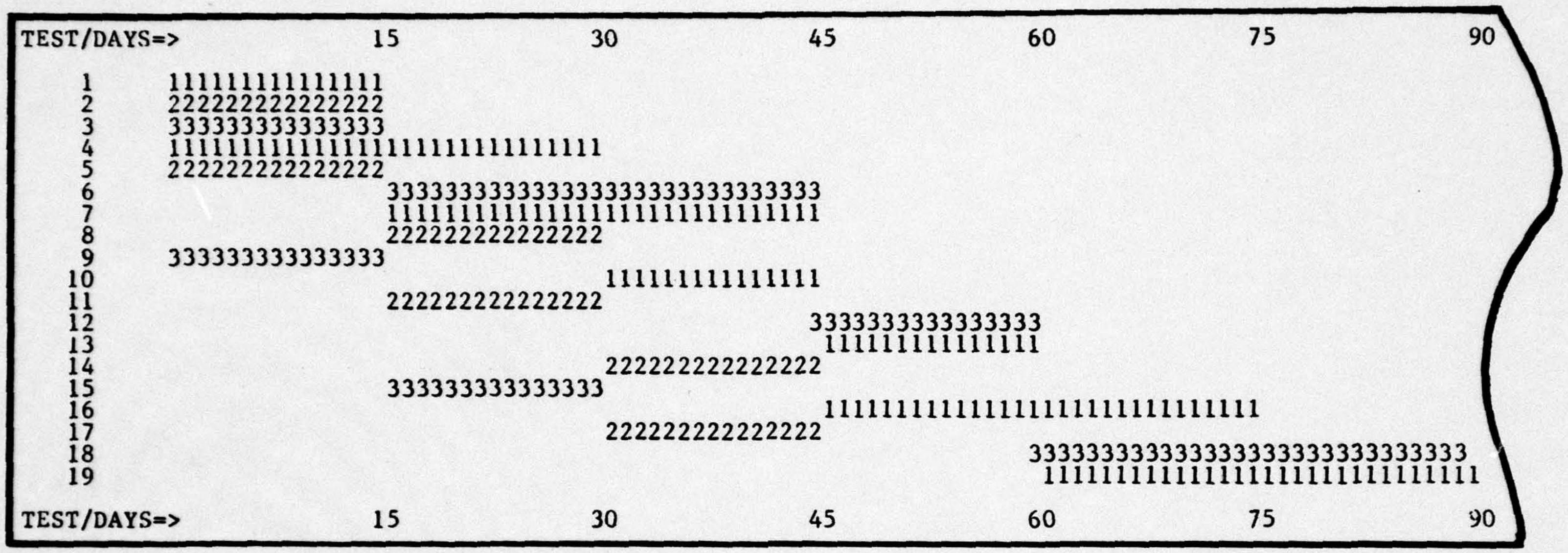
Additional runs were made to investigate the feasibility of using abbreviated runs to simulate long schedules. A glance at Figure 4, which is a fairly typical test-phasing bar graph, shows that there is considerable redundancy within spans of fifteen days. Days one through fifteen are identical, as are days sixteen through thirty. Except for days 46, 60 and

90 which are unique, there is a great deal of redundancy. The reason that certain days seem unique is that on the 45th day, for example, most of the subjects completed the last administration of a prerequisite test in the first block of the day, enabling them to be scheduled for milestone test 12 in the first afternoon period of the 45th day. A similar situation occurs on the 60th day.

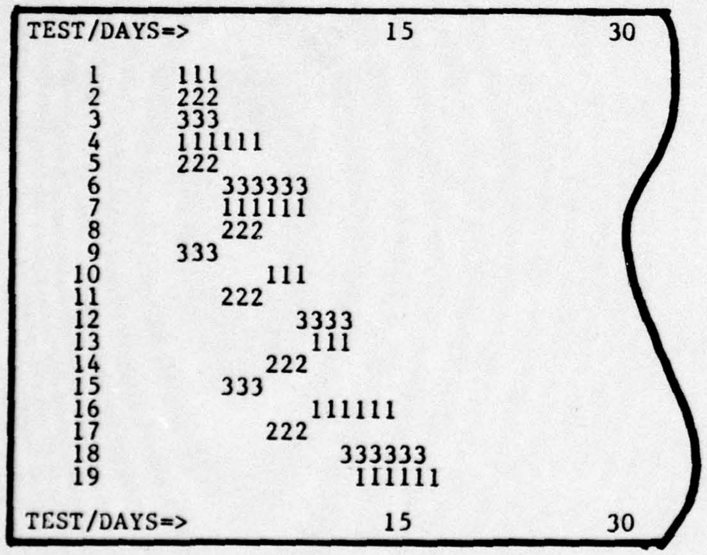
It seemed distinctly possible that the results of this run could have been obtained by appropriate extrapolation from another run, identical in every respect except that it required only three administrations of each test, or even only one administration. To test this possibility, three of the runs described in the preceding section were rerun with identical parameters except that the number of administrations required was reduced from fifteen to three. In addition, one of those was also run with one administration. In Figure 6, the runs which were repeated are the ones for ten subjects with eight periods per day, twenty subjects for ten periods, and fourteen subjects for ten periods.

The results of these reruns were as anticipated. The total lengths of the three administration schedules were one-fifth as long as their fifteen-administration counterparts, utilization factors were identical, and the test phasings were comparable (for individual subjects as well as tests as a whole). Figure 7 shows a comparison of the bar graphs for the three runs made using fourteen subjects and ten periods. The upper portion shows the fifteen-administration run which required 90 workdays, while the lower half shows the comparable three-administration and one-administration runs which took eighteen workdays and six workdays respectively. Note that, except for duration, the phasing is identical.

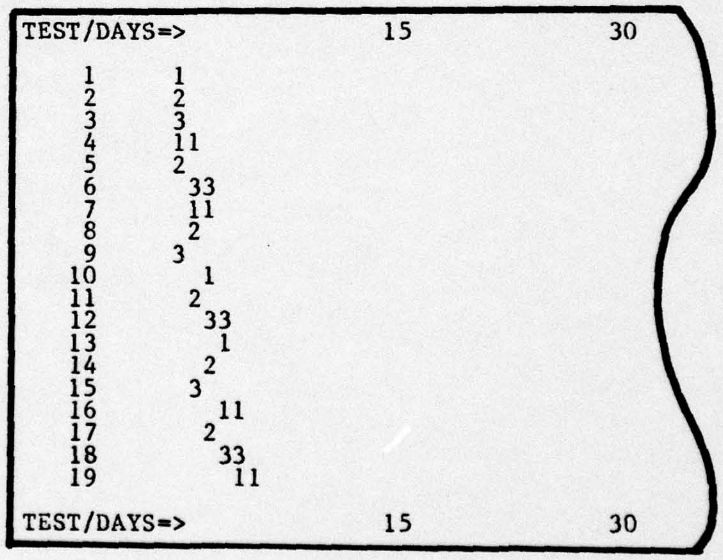
Whether valid extrapolation could be accomplished consistently in a run



```
TEST/DAYS=>            15             30             45             60             75             90

    1     111111111111111
    2     222222222222222
    3     333333333333333
    4     111111111111111111111111111111
    5     222222222222222
    6                    333333333333333333333333333333
    7                    111111111111111111111111111111
    8                    222222222222222
    9     333333333333333
   10                                   111111111111111
   11                    222222222222222
   12                                                  333333333333333
   13                                                  111111111111111
   14                                   222222222222222
   15                    333333333333333
   16                                                  111111111111111111111111111111
   17                                   222222222222222
   18                                                                 333333333333333333333333333333
   19                                                                 111111111111111111111111111111

TEST/DAYS=>            15             30             45             60             75             90
```



```
TEST/DAYS=>            15             30

    1     111
    2     222
    3     333
    4     111111
    5     222
    6        333333
    7        111111
    8        222
    9     333
   10           111
   11        222
   12             3333
   13              111
   14           222
   15        333
   16              111111
   17           222
   18               333333
   19                111111

TEST/DAYS=>            15             30
```



```
TEST/DAYS=>            15             30

    1     1
    2     2
    3     3
    4     11
    5     2
    6      33
    7      11
    8      2
    9     3
   10       1
   11      2
   12       33
   13        1
   14       2
   15      3
   16        11
   17       2
   18        33
   19         11

TEST/DAYS=>            15             30
```

Figure 7: Comparison of Phasing for Fifteen Versus Three Versus One Administration(s) for Ten Subjects and Eight Periods Per Day

requiring only one administration has not been established. However, it appears that the total length of a run is unaffected by occasional early starts, and is primarily determined by the manner in which bottleneck (individually-administered) tests can be handled under a given set of run conditions. In some instances these tests must be run end-to-end, while in other circumstances they can be doubled up or partially overlapped. In every instance seen to date, the total length of the schedule has turned out to be a multiple of the number of administrations required for each test. Thus it seems likely that this type of extrapolation would prove feasible, even when only one administration was required.

IV. CONCLUSIONS

Based on the runs described in the previous section, plus some preliminary runs made during program development, certain general conclusions are possible concerning the type of scheduling for which the SCHEDULER algorithm was designed. Individually-administered tests obviously constitute a bottleneck. Where there is a choice, group-administered tests are preferred, other considerations being equal. Where it is feasible to do so, providing duplicate facilities for individually-administered tests greatly reduces the schedule length.

Given the mix of test types and the number of tests within types employed in these runs, there are certain strategies which are superior. It seems reasonable to say that allowing small-group and group tests to be scheduled at less than capacity is a good strategy. Since all runs described here employed this rule it is not possible to show its superiority to a rule requiring full capacity to be maintained, but in any situation where some subjects are already assigned to individual tests it would not be possible to schedule a group test at the same time if a full capacity rule were to be enforced.

It is best to give first priority to individually-administered tests and last priority to group tests. Alternatives were found to be inferior. When group tests have priority they capture all the subjects, causing the other labs to sit idle.

Inefficiency results when it is not permitted to change tests once they have been set up for the day, because the labs become idle for lack of subjects after a few periods. It was found to improve efficiency if the day can be divided into two blocks of time periods. When the number of subjects is sufficiently small, blocking allows completion of two tests of each type per workday.

It was also observed that improved efficiency results when the scheduling is coordinated between two labs set up for the same type of test. There are times when two labs should, if possible, administer the same test. However, there are other times when it is more efficient to change over the second lab to another test when it is determined that the first lab has the capacity to carry the load alone.

The runs described here tend to indicate that the number of tests any subject can be expected to take is probably not excessive. As Figure 6 shows, the maximum number of administrations per subject per day was six. The highest average in any run was 5.81 per day.

Efficiency in terms of either the utilization factor or the average number of administrations per day is somewhat deceptive because it would be possible to add selected tests to the task which would fill up the schedule without increasing its length. The best efficiency measure would seem to be the number of workdays required to complete a given schedule.

V. SUGGESTED FUTURE DEVELOPMENT

The SCHEDULER program as it now exists must be considered a working and partially tested feasibility model which runs in a particular environment. Although written primarily in IBM FORTRAN IV language which implies a considerable degree of portability, it employs a few routines peculiar to the environment in which it was developed. Adapting the program for other installations would require some modification.

As a feasibility model, SCHEDULER does not achieve optimized allocation of subjects to tests. In this respect it is estimated that an improved version could suboptimize the allocation of subjects on a period by period basis by producing alternative schedules within each period and selecting the one which provides the greatest number of administrations per period. In essence, instead of using a fixed priority sequence for types of tests, it would select the best allocation within each period after trying various possibilities. Such a procedure can be expected to provide an improvement in the efficiency of the schedules produced, but at the cost of increased computer running time.

It may be possible to offset this increase in running time by further experimenting with the algorithm to develop techniques for breaking runs into phases. It would seem feasible to identify distinct phases in each simulation run, such as start-up, mid-period and ending phase. If the mid-phase is found in general to be a steady-state period, it should be possible to extrapolate from data obtained in a short sample of mid-period activity so that relatively short computer runs could be used to simulate relatively long scheduling tasks. This is based on the assumption that distinct phases exist in typical schedules and that they can be identified.

The runs made to date were designed to provide a demonstration of the major features of the algorithm, rather than being necessarily representative of real scheduling tasks in all respects. In particular, these runs employed a limited number of tests, whereas the NAMRL performance testing program is an on-going operation. The essential difference is that early in each experimental run of SCHEDULER the algorithm begins to run out of certain types of tests, so that a great portion of each run could be considered to be within the end phase. After 30 to 45 days, most of the group tests and small group tests have been completed and the majority of the remaining time is devoted to individual tests and those group tests which are also milestones. If there were more small group and group non-milestone tests to be scheduled, the steadily diminishing efficiency noted in most of the runs would be postponed until the very end of the run.

It is desirable that any extrapolation technique employed be readily applicable to the data obtained from a run. The labor required in the extrapolation should not outweigh the cost of the comparable full computer run. It is expected that the extrapolation could either be accomplished with a minimum of human effort or could be carried out automatically at the end of each computer run.

One problem anticipated in extrapolation from abbreviated runs is that they would have to be run strictly with respect to workdays, since holidays and leave days would be a complicating factor. Holidays perhaps could be taken into account in determining a specific end date, but leave days might constitute a more difficult problem, since the algorithm looks ahead to insure that no subject starts a test that may be interrupted by leave. It would be difficult to take such a procedure into account in extrapolated runs.

APPENDIX I: PROGRAM DESCRIPTION

SCHEDULER was written in IBM 360/370 FORTRAN IV language and was run under the IBM OS operating system using the RJE remote job entry system. The installation in which jobs were run allows extensive use of disk-stored card image input/output files. Advantage was taken of this capability by routing different types of output to separate files. Similarly, different types of input data were placed in separate disk files and read from separate units.

A. INPUT/OUTPUT FILES

Symbolic unit designation was employed to facilitate changing the physical unit designation if necessary, since units are assigned specific values in a BLOCK DATA Subroutine. These assignments follow a pattern. All units are designated by the letter U (defined as an integer) followed by one or two digits. All units ending in 5 are card reader-like devices specified as disk files having 80-character record lengths (of which only 72 are used). All units ending in 6 are printer-like output devices (defined as disk files) having 133-character line lengths, the first character of which is an IBM carriage control character. All units ending in 7 are card punch-like devices (defined as disk files) having 80-character record lengths. Figure A1 is a table of Input/Output files used in SCHEDULER.

Figure A2 gives the makeup of the input decks read by units U5 and U25, showing the sequence in which items are read. Among other things, the control cards must specify whether there is a probabilities card or not and the number

FILE DESIGNATORS, TYPES AND FUNCTIONS

SYMBOL	TYPE	FUNCTION
U5	READ	INPUT CONTROL CARDS, SUBJECTS & TESTS
U6	PRINT	OUTPUT DAILY SCHEDULE & BAR GRAPH
U7	PUNCH	OUTPUT LOGG ARRAY FOR RESTART (OPT.)
U16	PRINT	PRINT CALENDAR
U25	READ	INPUT CALENDAR SPECIAL DATES
U26	PRINT	PRINT DETAILED DIAGNOSTICS
U27	PUNCH	OUTPUT QUEUES, ETC. FOR RESTART (OPT.)
U35	READ	INPUT LOGG-ARRAY FOR RESTART (OPT.)
U36	PRINT	PRINT FINAL RESULTS (TEST START/STOPS, ETC.)
U37	PUNCH	OUTPUT DATA FOR PLOT OF % COMPLETED
U45	READ	INPUT QUEUES FOR RESTART
U47	PUNCH	OUTPUT LEGEND FOR PLOT TITLE
U57	PUNCH	PRINT SUBJECT START/STOP TABLE
U65	READ	INPUT START/STOPS FOR RESTART (BINARY)
U67	PUNCH	OUTPUT START/STOPS FOR RESTART (BINARY)

INPUT & OUTPUT FILES BY CLASS, UNIT NO. AND PROGRAM SEGMENT

CLASS→	CARD INPUT					PRINT				PUNCH					
UNIT→	5	25	35	45	65	6	16	26	36	7	27	37	47	57	67
PROGRAM SEGMENT															
MAIN	X	X				X	X	X	X			X	X	X	
DAYER						X		X			X				X
LSTLOG						X		X			X				X
CALEND							X								
LOGGER			X	X	X	X		X							
LOGOUT										X					
PLOTER									X				X		
FREEKY															
BARGRF						X									
BLOCK DATA															

Figure A1: SCHEDULER Input and Output Files

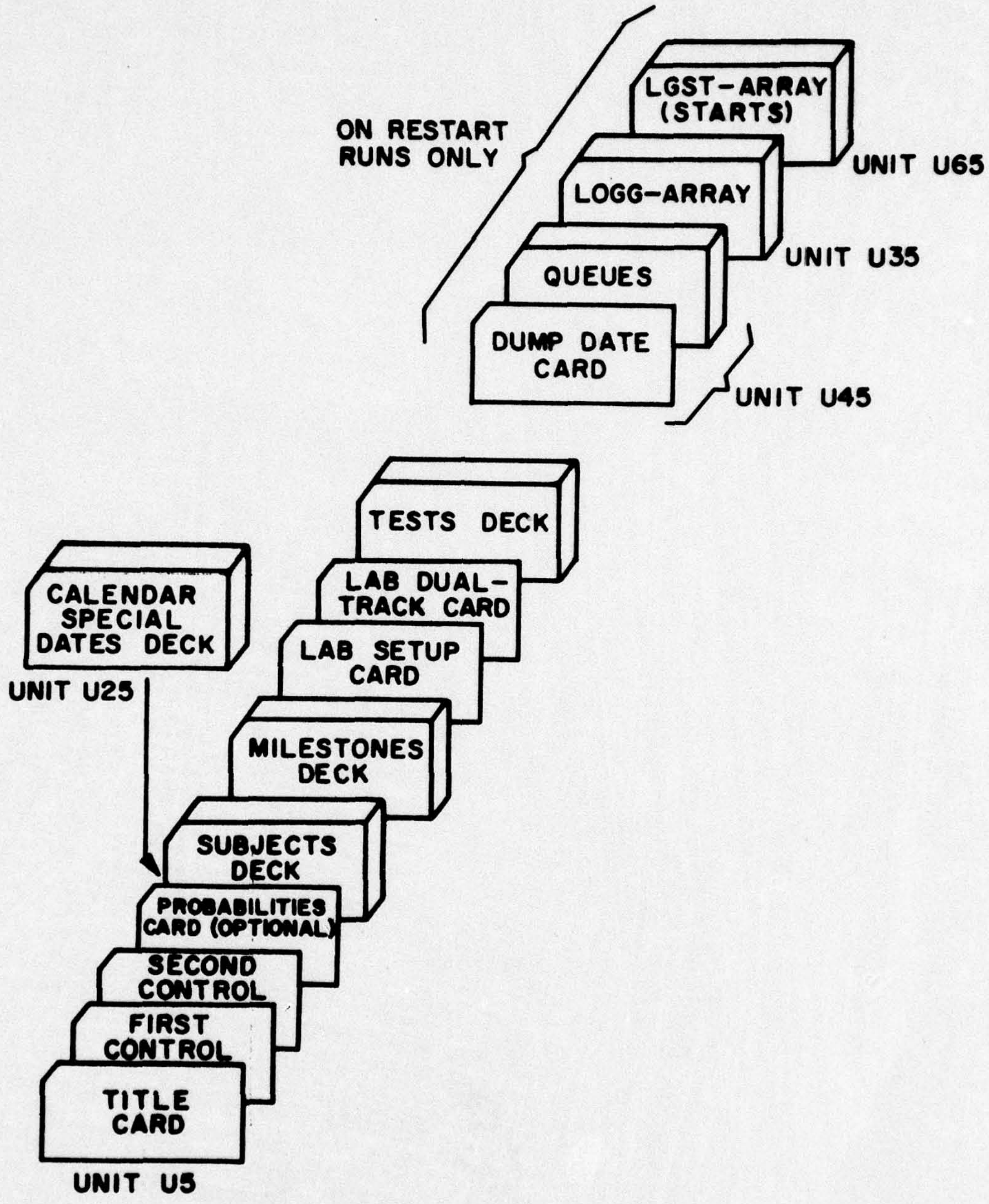


Figure A2: SCHEDULER Input Card Sequence

of subject cards and milestone cards included. Calendar cards, if any, are read from unit 25 before subject cards are read from unit 5. When an end-of-file is detected on unit 25, reading continues on unit 5, so it is not necessary to specify the number of date cards included in unit 25. All reading is done in the MAIN program, except for reading of restart files which is done in subroutine LOGGER.

Figures A3 through A11 show the field formats of the control cards and various data input cards. The title card (not shown) is free format. It allows up to 72 characters of title or other identification to be printed on all outputs.

B. INSTALLATION DEPENDENT CODE AND OTHER LIMITATIONS

SCHEDULER is known to present certain portability problems, particularly when attempting to convert to a UNIVAC machine. One of these is the use of symbolic device names in I/O statements. Others include the use of logical operators in LOGICAL IF statements and the use of literals enclosed in single quotes. Perhaps the most troublesome might be the extensive use of nested subscripts, as in

```
IN = J(K(L(M)))
```

which must be rewritten as

```
 N = L(M)
 I = K(N)
IN = J(I) .
```

In addition there are a number of installation-dependent routines employed in SCHEDULER, some of which are not essential and can be eliminated, such as the use of an interval timer. Others, such as the use of local

RECORD TITLE: CONTROL CARD NO. 1 UNIT: U5 FORMAT STATEMENT NO.: 20

FIELD NAME	FORMAT	POSITIONS	CONTENTS	MAX. VALUE
NWEEKS	I5	1-5	Length of calendar in weeks	208
NTESTS	I5	6-10	Number of tests	120
NSUBJS	I5	11-15	Number of subjects	100
NSLOTS	I5	16-20	Number of periods per day	---
NLABYS	I5	21-25	Number of laboratories	10
NADMIN	I5	26-30	No. of admins. required per test: Defaults to 15 if blank or zero If $>$ 0, provides default for test cards If $<$ 0, overrides values on test cards	---
NORANGE	I5	31-35	Number of tests allowed active at once	100
NLINKS	I5	36-40	Number of tests which are "milestones"	20
MINGSZ	I5	41-45	"Min. group size"-option switch: If $\leq$ 0, option not used If $\geq$ 0, option is used	---
MAXTST	I5	46-50	Max. No. of tests per day anyone may take	---
MAVAIL	I5	51-55	Subject "unavailability" simulation switch: If $>$ 0, subject unavailability simulated If $\leq$ 0, option not used	---
MOUTAG	I5	56-60	Lab. "outage" simulation switch: If $>$ 0, lab outages simulated If $\leq$ 0, option not used	---
LSTLOW	I5	61-65	Listing level to be used after a Dump	---
MCPRNT	I5	66-70	Controls calendar printing: If $\leq$ 0, calendar not printed If $\geq$ 0, MCPRNT weeks are printed	208

Figure A3: Record Format for Control Card No. 1

RECORD TITLE: CONTROL CARD NO. 2 UNIT: U5 FORMAT STATEMENT NO.: 22

FIELD NAME	FORMAT	POSITIONS	CONTENTS	MAX. VALUE
START	8A1	1-8	Calendar starting date (MM/DD/YY)	---
PHYRST	8A1	11-18	First date of testing (MM/DD/YY) Must be a Monday Defaults to first Monday after START	---
LASTWK	I5	21-25	Max. weeks this run may take	208
NEWDAY	I5	26-30	First workday of week (2=Mon.) Defaults to 2 (Monday)	7
LASTDY	I5	31-35	Last workday of week (6=Fri.) Defaults to 6 (Friday)	7
NORULE	I5	36-40	Switch controls unrestricted scheduling: If = 0, option not used If ≠ 0, tests may be changed anytime in any lab	---
LOGGIN	I5	41-45	Controls reading dumped files: If ≤ 0, dumped files not read If > 0, this is a restart run	---
LOGDMP	I5	46-50	Controls writing dump files for restart If ≤ 0, files are not dumped If > 0, files will be dumped at end of the LOGDMPth week	208
IBLOCK	I5	51-55	Controls splitting day into 2 blocks of periods: If ≤ 0, no blocking takes place If > 0, 2nd block starts in 1BLOCKth period	---
MINGO	I5	56-60	Controls writing details to Unit U26 Specifies week when detail increases (Before MINGO, detail is minimal; After MINGO it greatly increases)	208
LSTLVL	I5	61-65	Controls level of detail written to Unit U6 Ten levels are possible. Low LSTLVL gives least detail. LSTLVL of 4 or more prints daily schedule with names.	---
NRACRS	I2	66-70	Controls no. of names listed on each line. Maximum is 10. Default is 10.	---

Note: See Figure A6 note on date formats

Figure A4: Record Format for Contrcl Card No. 2

RECORD TITLE: PROBABILITIES CARD   UNIT: U5   FORMAT STATEMENT NO.: 260

FIELD NAME	FORMAT	POSITIONS	CONTENTS	MAX. VALUE
PRUNAV	F5.3	1-5	Probability that a subject is temporarily unavailable (for one day)	1.000
IBASE	I10	11-20	Seed for starting random number generator. Default value is provided by subroutine if no user seed is given.	$2^{31}-1$
PROUTS(I)	10F6.3	21-75	Probability that lab has temporary outage (one value for each of up to 10 labs)	1.000

Figure A5: Record Format for Probabilities Card

RECORD TITLE: CALENDAR SPECIAL DATE CARD UNIT: U25

FORMAT STATEMENT NO.: 305

FIELD NAME	FORMAT	POSITIONS	COMMENTS	MAX. VALUE
GDATE	8A1	1-8	Reference date for positioning special dates specified. (MM/DD/YY) Must be in calendar range.	---
MINCOD	A2	9-10	Code designating type of format: If blank, card is type "A" If '--', card is type "B"	---
JCODE(I)	7I1	11-17	Each position corresponds to a day of the week (2=Monday) Coded entries specify type of special date: 1 = holiday, 2 = blue team leave day, 3 = gold team leave day	---

Notes:

1. If card is type "A" (MINCOD is blank):

   a. GDATE must be a Sunday and is first day of week in which designated special date(s) fall.

   b. JCODE array elements which are not blank are special dates. Corresponding elements of calendar array are loaded with holiday or leave codes as specified. Other days of that week are set blank, replacing values set by previously read card for same week, if any.

2. If card is type "B" (MINCOD is '--'):

   a. GDATE is the date of one specific holiday (12/25/80, for example).

   b. Day of the week need not be determined.

   c. Only one date may be specified per card.

   d. Code to be entered in calendar must be specified in JCODE(1) position 11.

   e. Other days of that week are unaffected.

3. Actually, dates may be in any of 3 formats: YY Julian day, DD/MM/YY or MM/DD/YY. Program converts all to MM/DD/YY.

Figure A6: Record Format for Special Date Cards

RECORD TITLE: SUBJECT CARD UNIT: U5 FORMAT STATEMENT NO.: 550

FIELD NAME	FORMAT	POSITIONS	CONTENTS	MAX. VALUE
SUBNAM(J,I)	8A1	1-8	Subject name or I.D. (anything)	---
ITEAM	I2	11-12	Leave team assignment 1 = Blue 2 = Gold If = 0, will be randomly assigned	2
SENTRY	8A1	15-22	Entry date (MM/DD/YY)	---
SEXITS	8A1	25-32	Exit date (MM/DD/YY)	---

Note: See Figure A6 note on date formats

Figure A7: Record Format for Subject Cards

RECORD TITLE: MILESTONE TEST CARD UNIT: U5 FORMAT STATEMENT NO.: 705

FIELD NAME	FORMAT	POSITIONS	CONTENTS	MAX. VALUE
J	I5	1-5	Milestone Number (1, 2, 3,...)	20
MILK	I5	6-10	Test Number (6, 12, 18...) (A "Backpointer" not used by program)	---
LLS(J)	I5	11-15	Number of items in this list of prerequisites	10
LINK(J,I)	10I5	16-65	List of prerequisite tests	120

Figure A8: Record Format for Milestone Cards

RECORD TITLE: LABORATORY SETUP CARD UNIT: U5 FORMAT STATEMENT NO.: 760

FIELD NAME	FORMAT	POSITIONS	CONTENTS	MAX. VALUE
LABSET(K)	10I5	1-50	Type of test each lab is set up for, one entry per lab. (Values are normally positive; user may set any negative to suspend for that lab the rule which prohibits changing tests except in first period of day or first period of second block.)	---

Figure A9: Record Format for Laboratory Setup Card

RECORD TITLE: LABORATORY DUAL-TRACK DESIGNATOR CARD UNIT: U5

FORMAT STATEMENT NO.: 760

FIELD NAME	FORMAT	POSITIONS	CONTENTS	MAX. VALUE
LABKEY(K)	10I5	1-50	Lab. number of first track of a dual-track pair. This entry is placed in the array element corresponding to the second track of the dual-track pair. (e.g., if labs 1 and 2 are a dual-track pair, enter a 1 in LABKEY(2) to show that lab 2 is the second track of a pair of which lab 1 is the first.	10

Figure A10: Record Format for Laboratory Dual-Track Card

RECORD TITLE: TEST CARD UNIT: U5 FORMAT STATEMENT NO.: 805

FIELD NAME	FORMAT	POSITIONS	CONTENTS	MAX. VALUE
J	I5	1-5	Test number (must be from a continuous sequence of integers)	120
MTYPES(J)	I5	6-10	Test type code (1 = individual, 2 = small group, 3 = group)	3
MAXGRP(J)	I5	11-15	Max. number of subjects accommodated (Enter 101 if whole group)	100
MINGRP(J)	I5	16-20	Min. No. of subjects accommodated	---
MADMIN(J)	I5	21-25	No. of administrations required (Defaults to 'NADMIN' on control card)	---
MILEST(J)	I5	26-30	Pointer to milestone list (Blank if test is not a milestone, else specify which list)	20
KSTART	I5	31-35	Week when test is to start (Blank if test may start immediately)	208
KSTOPT	I5	36-40	Week in which test must stop (Blank if test has no cutoff)	208
NSTART	8A1	42-50	Date test may start (MM/DD/YY)	---
NSTOPT	8A1	52-60	Date test must stop (MM/DD/YY)	---

Figure All: Record Format for Test Cards string manipulation routines like CHMOVE and the IBM absolute value function IABS, may have UNIVAC equivalents. The use of FILL as a fast means of initializing arrays can be replaced by conventional DO-loops.

Extensive use is made in SCHEDULER of two Julian date conversion routines DATEJ and JDAY. It may be possible to find equivalent UNIVAC routines, but their elimination would have a serious impact on program operation. While Julian routines would not be necessary if scheduling were done strictly with respect to workdays, eliminating the calendar feature would require extensive reprogramming. The source code for these two routines is available, but unfortunately they in turn call other local routines, some of which are written in IBM assembler language, thus presenting further conversion problems.

The percentage completion graphs described in Section II.A. of this report were produced by a separate plotting program which processes the X-Y data output by SCHEDULER to units U37, and the legend information in unit U47. This program, called GRAPHAG, converts the X-Y pair data to a file of pen codes which can be used by AGILE terminal equipment in an X-Y plotting mode to produce the desired plots of percentage completion as a function of time. Similar UNIVAC routines may exist which can produce plots from a set of X-Y pairs, perhaps on a line printer.

C. PROGRAM RUN TIME

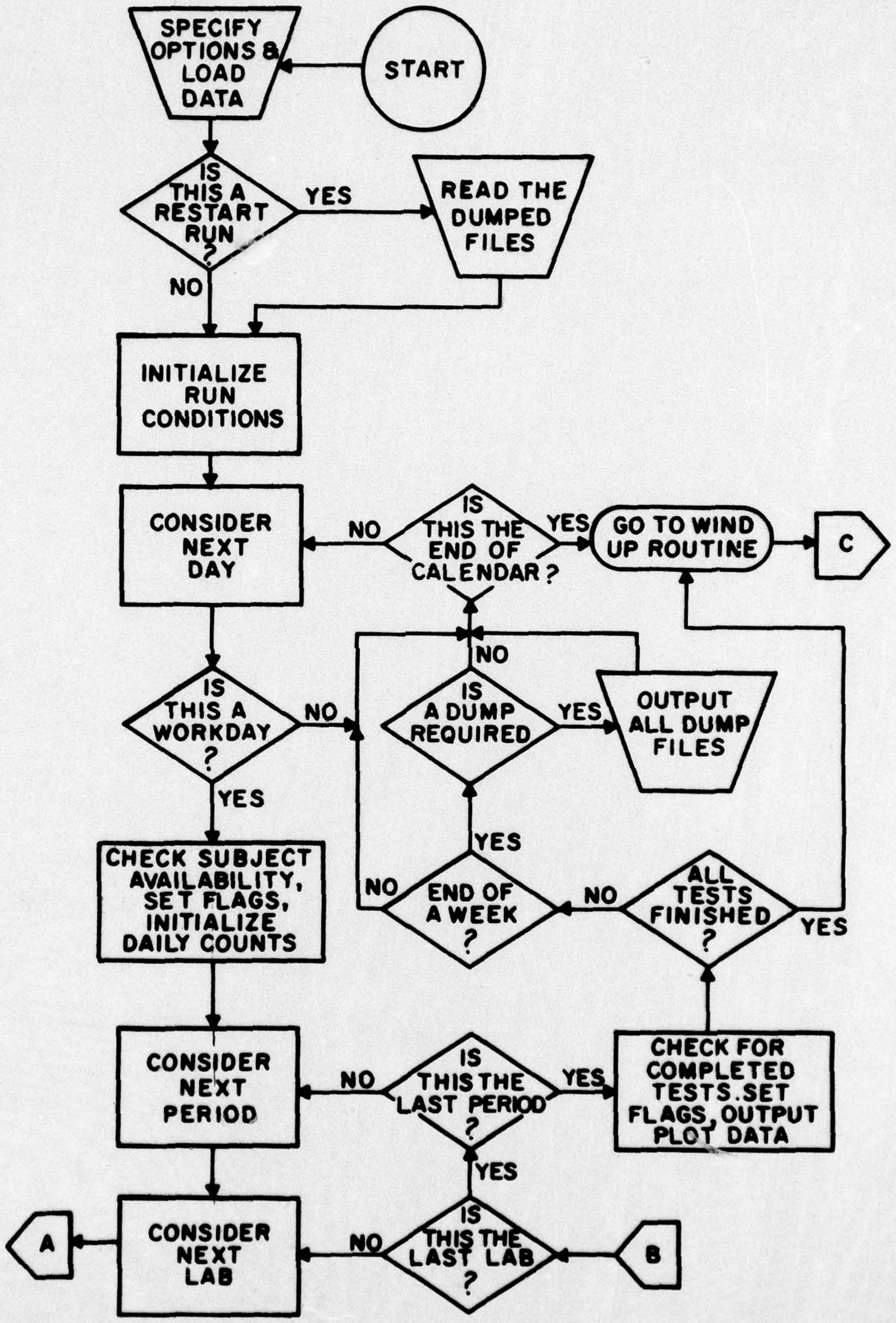
Typical run times (on an IBM 370 computer with 3033 CPU using MVT) for nineteen tests, fourteen subjects, four labs and ten periods per day are:

Administrations	Time (seconds)
15	23
3	8
1	7

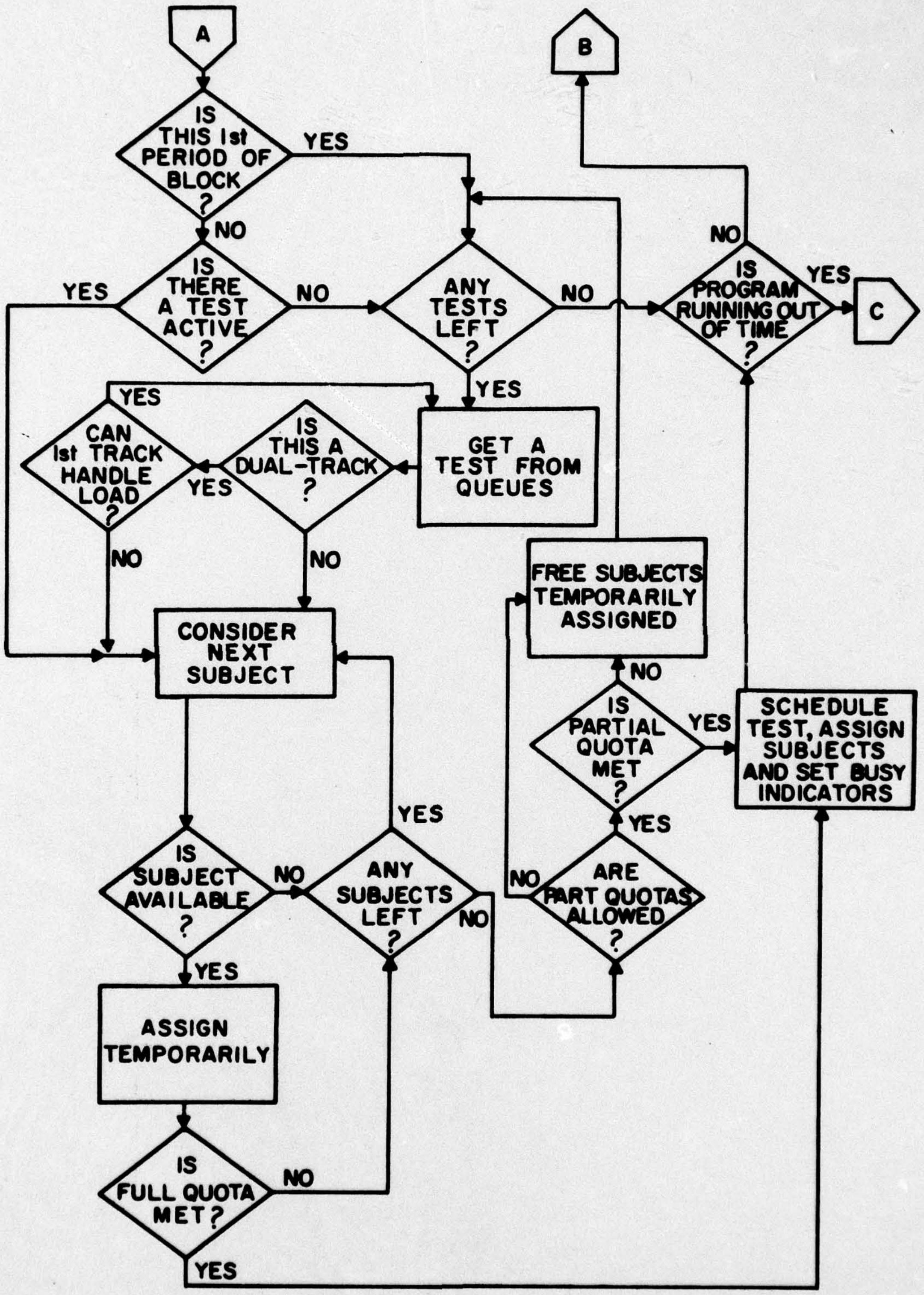
These run times include compilation and system overhead time charges as well as some fixed time usage involved in initialization and windup within SCHEDULER.

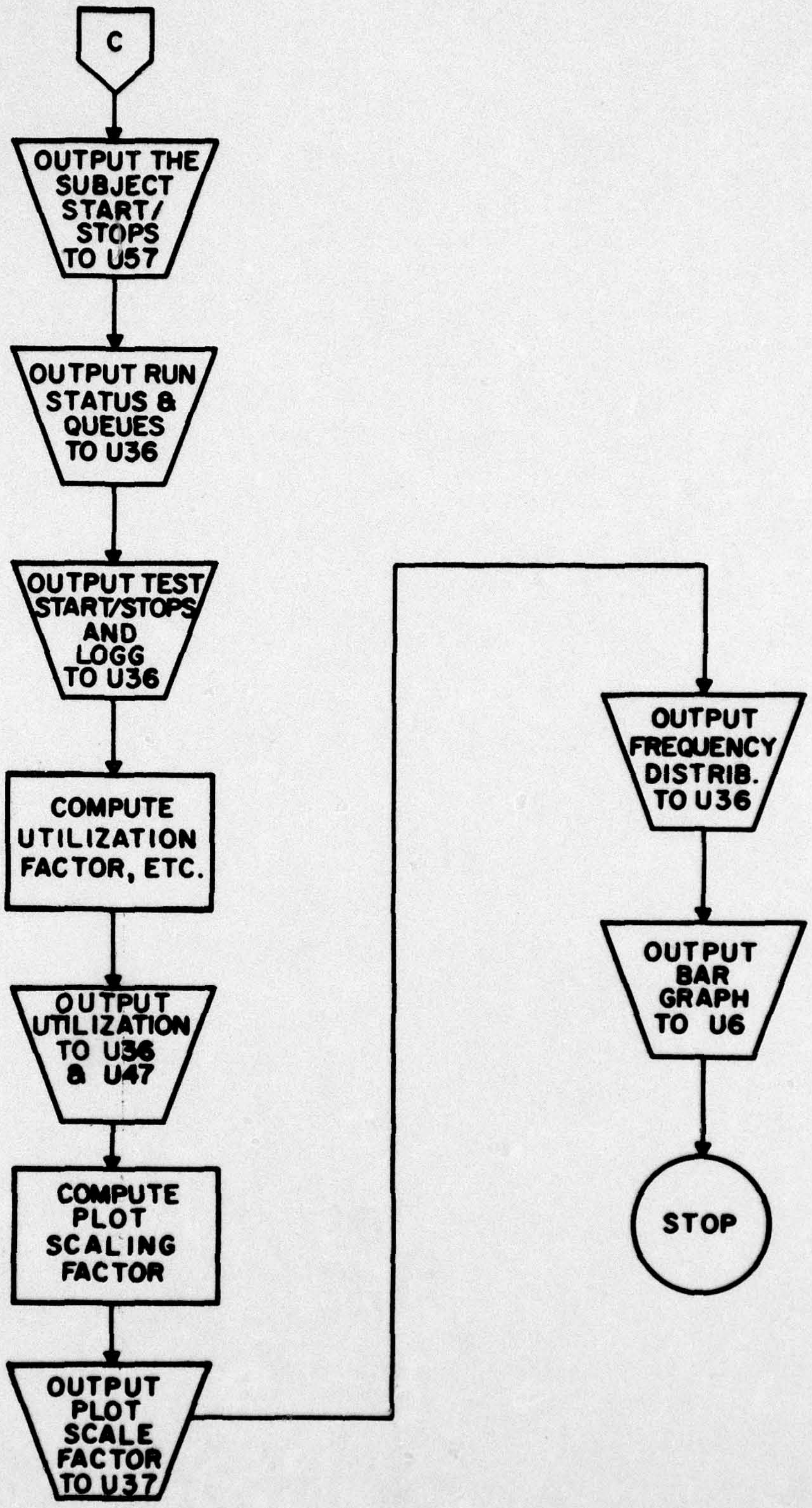
APPENDIX II: FLOW CHARTS

The following pages present a system flow chart designed to show the broad general logic of the SCHEDULER program.



SPECIFY OPTIONS & LOAD DATA
START
IS THIS A RESTART RUN ?
YES
READ THE DUMPED FILES
NO
INITIALIZE RUN CONDITIONS
CONSIDER NEXT DAY
NO
IS THIS THE END OF CALENDAR ?
YES
GO TO WIND UP ROUTINE
C
NO
IS THIS A WORKDAY ?
NO
IS A DUMP REQUIRED
YES
OUTPUT ALL DUMP FILES
YES
YES
CHECK SUBJECT AVAILABILITY, SET FLAGS, INITIALIZE DAILY COUNTS
NO
END OF A WEEK ?
NO
ALL TESTS FINISHED ?
YES
CONSIDER NEXT PERIOD
NO
IS THIS THE LAST PERIOD ?
YES
CHECK FOR COMPLETED TESTS. SET FLAGS, OUTPUT PLOT DATA
YES
A
CONSIDER NEXT LAB
NO
IS THIS THE LAST LAB ?
B


A
IS THIS 1st PERIOD OF BLOCK ?
YES
NO
IS THERE A TEST ACTIVE ?
YES
NO
ANY TESTS LEFT ?
NO
YES
B
NO
IS PROGRAM RUNNING OUT OF TIME ?
YES
C
YES
CAN 1st TRACK HANDLE LOAD ?
YES
IS THIS A DUAL-TRACK ?
GET A TEST FROM QUEUES
NO
NO
FREE SUBJECTS TEMPORARILY ASSIGNED
CONSIDER NEXT SUBJECT
NO
IS PARTIAL QUOTA MET ?
YES
SCHEDULE TEST, ASSIGN SUBJECTS AND SET BUSY INDICATORS
YES
YES
IS SUBJECT AVAILABLE ?
NO
ANY SUBJECTS LEFT ?
NO
NO
ARE PART QUOTAS ALLOWED ?
YES
ASSIGN TEMPORARILY
IS FULL QUOTA MET ?
NO
YES


C
OUTPUT THE SUBJECT START/ STOPS TO U57
OUTPUT RUN STATUS & QUEUES TO U36
OUTPUT TEST START/STOPS AND LOGG TO U36
COMPUTE UTILIZATION FACTOR, ETC.
OUTPUT UTILIZATION TO U36 & U47
COMPUTE PLOT SCALING FACTOR
OUTPUT PLOT SCALE FACTOR TO U37
OUTPUT FREQUENCY DISTRIB. TO U36
OUTPUT BAR GRAPH TO U6
STOP

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 112-3	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (*and Subtitle*) SCHEDULER: A COMPUTER PROGRAM FOR SCHEDULING ADMINISTRATION OF PERFORMANCE TESTS		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robert L. Gardner and Dennis E. Smith		8. CONTRACT OR GRANT NUMBER(s) N00014-79-C-0128
9. PERFORMING ORGANIZATION NAME AND ADDRESS Desmatics, Inc. P. O. Box 618 State College, PA 16801	391 152	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 207-037
11. CONTROLLING OFFICE NAME AND ADDRESS Biophysics Program (Code 444) Office of Naval Research Arlington, VA 22217		12. REPORT DATE November 1979
		13. NUMBER OF PAGES 52
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)		15. SECURITY CLASS. (*of this report*) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Distribution of this report is unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

Scheduling Algorithm
Scheduling Administration
Computer Program
Performance Tests

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

This report describes SCHEDULER, a computer program which implements a heuristic scheduling algorithm. The algorithm is designed to produce a feasible schedule for administration of performance tests to a group of subjects who are required to take multiple administrations of each of several tests. Provision is made for scheduling up to ten tests in parallel during each of several periods per day. Availability of subjects and tests may be specified in terms of an internal calendar which may be linked to the real-world calendar, so that a completion date for the schedule may be determined.

The SCHEDULER program may be used experimentally in evolving scheduling strategies or may be used to produce a working schedule for a real set of circumstances. The results of a series of experimental runs are presented, as well as some conclusions regarding strategies for a specific application. The program input and output formats are described in detail and a system flow chart is provided.